Transfer Learning by Structural Analogy

Huayan Wang
Computer Science Department
Stanford University
353 Serra Street, Stanford, CA, U.S.A.

Qiang Yang
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Clearwater Bay, Kowloon, Hong Kong

Abstract

Transfer learning allows knowledge to be extracted from auxiliary domains and be used to enhance learning in a target domain. For transfer learning to be successful, it is critical to find the similarity between auxiliary and target domains, even when such mappings are not obvious. In this paper, we present a novel algorithm for finding the structural similarity between two domains, to enable transfer learning at a structured knowledge level. In particular, we address the problem of how to learn a non-trivial structural similarity mapping between two different domains when they are completely different on the representation level. This problem is challenging because we cannot directly compare features across domains. Our algorithm extracts the structural features within each domain and then maps the features into the Reproducing Kernel Hilbert Space (RKHS), such that the "structural dependencies" of features across domains can be estimated by kernel matrices of the features within each domain. By treating the analogues from both domains as equivalent, we can transfer knowledge to achieve a better understanding of the domains and improved performance for learning. We validate our approach on synthetic and real-world datasets.

Introduction and Motivation

Re-using knowledge across different learning tasks (domains) has long been addressed in the machine learning literature (Thrun 1998; Caruana 1997; Daumé III 2006; Dai 2008; Blitzer 2006). Existing research on this issue usually assume that the tasks are related on the *low* level, i.e. they share the same feature space or the same parametric family of models, such that knowledge transfer can be achieved by re-using weighted samples across tasks, finding a shared intermediate representation, or learning constraints (informative priors) on the model parameters.

However, examining knowledge transfer in human intelligence, we could find that human beings do not rely on such low-level relatedness to transfer knowledge across domains. Namely, we human beings are able to make analogy across different domains by resolving the *high* level (structural) similarities even when the learning tasks (domains) are seemingly irrelevant. For example, we can easily understand the analogy between debugging for computer viruses and diagnosing human diseases. Even though the computer viruses (harmful codes) themselves have nothing in common with bacteria or germs, and the computer systems is totally different from our bodies, we can still make the analogy base on the following *structural* similarities:

1. Computer viruses cause malfunction of computers. Diseases cause disfunction of the human body.
2. Computer viruses spread among computers through the networks. Infectious diseases spread among people through various interactions.
3. System updates help computers avoid certain viruses. Vaccines help human beings avoid certain diseases.

Understanding of these structural similarities helps us abstract away the details specific to the domains, and build a mapping between the abstractions (see Figure 1). The mapping builds on the high level structural relatedness of the two domains, instead of their low level "literal similarities". In other words, the attributes of the "computer" and the "human" themselves do not matter to the mapping, whereas their relationships to other entities in their own domains matter.

This is reminiscent of the *learning-by-analogy* paradigm in early endeavors in intelligent planing and problem solving. However, many previous operational systems in computational analogy, such as case-based reasoning, have used a simple similarity function between an old and new problem domain, whereby the features in the two domains are identical, albeit weighted. This similarity measure cannot handle some more intuitive cases of human problem solving, such as the above example, in which the similarity between the domains should be measured on the structural level. And such a "structural similarity" can only be determined if we can correctly identify *analogues* across completely different representation spaces.

On the other hand, in cognitive science, analogical learning indeed involves developing a set of mappings between features from different domains. Such a need is captured in structure mapping theory (Falkenhainer 1989; Gentner 1990) of analogical reasoning, which argued for deep relational similarity rather than superficial similarity. However, an operational computational theory has been lacking for

Copyright © 2011, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

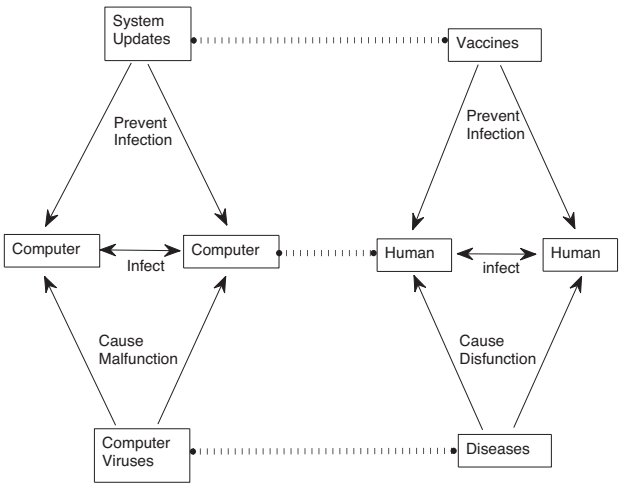

Figure 1: We can make the analogy between debugging for computer viruses and diagnosing human diseases based on structural similarities. The dash lines bridge analogues across domains.

how to come up with the mapping function. We try to fill this gap in this paper.

In this paper, we present a framework of *transfer learning by structural analogy*, which builds on functional space embedding of distributions (Smola 2007). Specifically, we address transfer learning in a setting that the source domain and target domain are using *completely different representation spaces*. As we cannot directly compare features across domains, we extract the structural information of the features within each domain by mapping the features into the Reproducing Kernel Hilbert Space (RKHS), such that the "structural dependencies" of features across domains can be estimated by kernel matrices of the features within each domain (Smola 2007). Hence the learning process is formulated as simultaneously selecting and associating features from both domains to maximize the dependencies between the selected features and response variables (labels), as well as between the selected features from both domains. With the learned cross-domain mapping, a structural similarity between the two domains can be readily computed, which can be used in place of simple similarity measures in computational analogy systems such as case based reasoning. By treating the analogues from both domains as equivalent, we can transfer knowledge to achieve a better understanding of the domains, e.g. better accuracy in classification tasks.

Related Work

The idea of re-using knowledge across learning tasks (domains) has been addressed in the machine learning literature in different terminologies, such as learning to learn, multi-task learning, domain adaptation, and transfer learning (Thrun 1998; Caruana 1997; Daumé III 2006; Dai 2008; Blitzer 2006; Mahmud 2007). To the best of our knowledge, among these works (Dai 2008) and (Mahmud 2007) are the only ones that address transferring knowledge across different representations spaces. However, (Dai 2008) rely on co-occurrence observations that bridges the two feature spaces (such as a dictionary, which consists of co-occurrence observations of two languages), such that the cross-domain relations of the features can be estimated straightforwardly. In contrast, our work does not rely on the availability of such co-occurrence data. (Mahmud 2007) proposed theoretical foundations for transfer learning between arbitrary tasks based on Kolmogorov complexity. However they only showed how to implement their framework in the context of decision trees, whereas our framework of making structural analogy between the features can be applied together with many different learning algorithms.

Learning by analogy is one of the fundamental insights of artificial intelligence. Humans can draw on the past experience to solve current problems very well. In AI, there has been several early works on analogical reasoning, such as Dynamic Memory (Schank 1982). Using analogy in problem solving, (Carbonell 1981; Winston 1980) pointed out that analogical reasoning implies that the relationship between entities must be compared, not just the entity themselves, to allow effective recall of previous experiences. (Forbus 1998) has argued for high-level structural similarity as a basis of analogical reasoning. (Holyoak 1997) has developed a computational theory of analogical reasoning using this strategy, when abstraction rules given as input that allow the two instances to be mapped to a unified representation.

Analogical problem solving is the cornerstone for case-based reasoning (CBR), where many systems have been developed. For example, HYPO (Ashley 1991) retrieves similar past cases in a legal case base to argue in support of a claim or make counter-arguments. PRODIGY (Carbonell 1991) uses a collection of previous problem solving cases as a case base, and retrieves the most similar cases for adaptation.

However, most operational systems of analogical reasoning, such as CBR systems (Aamodt 1994; Watson 1997; Leake 1996; Kolodner 1993), have relied on the assumption the past instances and the new target problem be in the same representational space. Most applications of CBR fall in this case (Mark 1989; Cheetham 2007; Bayoudh 2007), where the sets of feature that describe the old cases and new problems are the same. For example, cases for car diagnosis are built on descriptions of automobile attributes such as battery and engine size, although the values are allowed to be different between a past case and the current problem.

Approach

Estimating Structural Dependencies by HSIC

We aim at resolving the structural analogy between two domains with completely different low-level representations. For the source domain we are provided with observations

and response variables (labels):

$$\mathbb{S} = \{(x_1^{(s)}, y_1^{(s)}), (x_2^{(s)}, y_2^{(s)}), \cdots, (x_{N_s}^{(s)}, y_{N_s}^{(s)})\} \subset \mathcal{X}_s \times \mathcal{Y}_s, \quad (1)$$

where $\mathcal{X}_s$ is the source input domain and $\mathcal{Y}_s$ is the source output (label) domain. Similarity we have data for the target domain:

$$\mathbb{T} = \{(x_1^{(t)}, y_1^{(t)}), (x_2^{(t)}, y_2^{(t)}), \cdots, (x_{N_t}^{(t)}, y_{N_t}^{(t)})\} \subset \mathcal{X}_t \times \mathcal{Y}_t. \quad (2)$$

Note that $\mathcal{X}_t$, $\mathcal{Y}_t$ can be representation spaces that are *completely different* from $\mathcal{X}_s$, $\mathcal{Y}_s$.

For both the source and the target domain, we denote their feature domains as Φ_s and Φ_t. In practice, features are represented by their profiles[1] in the training set:

$$\{f_1^{(s)}, f_2^{(s)}, \cdots, f_S^{(s)}\} \subset \Phi_s, \quad (3)$$

$$\{f_1^{(t)}, f_2^{(t)}, \cdots, f_T^{(t)}\} \subset \Phi_t. \quad (4)$$

For vector representations, $(f_1^{(s)}, f_2^{(s)}, \cdots, f_S^{(s)})$ is simply the transpose of $(x_1^{(s)}, x_2^{(s)}, \cdots, x_N^{(s)})$. Nevertheless, our framework is applicable to more sophisticated representations (such as graphs *etc.*) as it is kernelized, which accesses data only through the kernel function.

Let $\mathcal{H}_s$, $\mathcal{H}_t$, $\mathcal{G}_s$, $\mathcal{G}_t$, $\mathcal{F}_s$, and $\mathcal{F}_t$ be reproducing kernel Hilbert spaces (RKHS) on the domains $\mathcal{X}_s$, $\mathcal{X}_t$, $\mathcal{Y}_s$, $\mathcal{Y}_t$, Φ_s and Φ_t, with associated kernel functions m_s, m_t, l_s, l_t, k_s and k_t respectively. Then we are able to estimate dependencies across domains using the Hilbert-Schmidt Independence Criterion (HSIC) (Gretton 2005; 2007; Smola 2007), which is defined as the square of the Hilbert-Schmidt norm of the cross-covariance operator bridging the two RKHS.

Specifically, for the RKHS $\mathcal{F}_s$ and $\mathcal{F}_t$ on the feature domains Φ_s and Φ_t, in terms of the kernel functions k_s, k_t the HSIC can be expressed as

$$\begin{aligned}\mathcal{D}(\mathcal{F}_s, \mathcal{F}_t, \Pr_{st}) = \mathbf{E}_{ss'tt'}[k_s(s,s')k_t(t,t')] \\ +\mathbf{E}_{ss'}[k_s(s,s')]\mathbf{E}_{tt'}[k_t(t,t')] \\ -2\mathbf{E}_{st}[\mathbf{E}_{x'}[k_s(s,s')]\mathbf{E}_{y'}[k_t(t,t')]],\end{aligned} \quad (5)$$

where $\Pr_{st}$ is the joint distribution of source and target domain features over $\Phi_s \times \Phi_t$, and (s,t), (s',t') are distributed independently according to the joint distribution.

Given a sample

$$\mathbb{F} = \{(f_1^{(s)}, f_1^{(t)}), (f_2^{(s)}, f_2^{(t)}), \cdots, (f_W^{(s)}, f_W^{(t)})\} \quad (6)$$

of the joint distribution $\Pr_{st}$, HSIC can be estimated using the kernel matrices (Song 2007):

$$\begin{aligned}\mathcal{D}(\mathcal{F}_s, \mathcal{F}_t, \mathbb{F}) = \frac{1}{W(W-3)}[\mathrm{tr}(\mathbf{K}^s\mathbf{K}^t) \\ +\frac{\mathbf{1}^\top\mathbf{K}^s\mathbf{1}\mathbf{1}^\top\mathbf{K}^t\mathbf{1}}{(W-1)(W-2)} - \frac{2}{W-2}\mathbf{1}^\top\mathbf{K}^s\mathbf{K}^t\mathbf{1}],\end{aligned} \quad (7)$$

where $\mathbf{K}^s(i,j) = (1-\delta_{ij})k_s(f_i^{(s)}, f_j^{(s)})$ and $\mathbf{K}^t(i,j) = (1-\delta_{ij})k_t(f_i^{(t)}, f_j^{(t)})$ are the kernel matrices with diagonal entries set to zero.

[1]The "profile" of a feature is defined as its feature value on all instances of a dataset.

Similarly, we can estimate the dependencies across the domains $(\mathcal{X}_s, \mathcal{Y}_s)$ and $(\mathcal{X}_t, \mathcal{Y}_t)$ by the corresponding kernel matrices $\mathbf{M}^s$, $\mathbf{L}^s$, $\mathbf{M}^t$ and $\mathbf{L}^t$ computed by the samples $\mathbb{S}$, $\mathbb{T}$ (in (1) and (2)) from the joint distributions $\Pr_{xy}^{(s)}$ and $\Pr_{xy}^{(t)}$, where $\mathbf{M}^s(i,j) = (1-\delta_{ij})m_s(x_i^{(s)}, x_j^{(s)})$, $\mathbf{L}^s(i,j) = (1-\delta_{ij})l_s(y_i^{(s)}, y_j^{(s)})$, $\mathbf{M}^t(i,j) = (1-\delta_{ij})m_t(x_i^{(t)}, x_j^{(t)})$ and $\mathbf{L}^t(i,j) = (1-\delta_{ij})l_t(y_i^{(t)}, y_j^{(t)})$.

Estimating dependencies by HSIC is a crucial component in our learning framework, which requires estimating dependencies for the three pairs of domains, namely the source input and output domain $(\mathcal{X}_s, \mathcal{Y}_s)$, the target input and output domain $(\mathcal{X}_t, \mathcal{Y}_t)$, and the source and target feature domain (Φ_s, Φ_t)

Transfer Learning by Structural Analogy

The joint distributions $\Pr_{xy}^{(s)}$ and $\Pr_{xy}^{(t)}$ are well characterized by the samples $\mathbb{S}$ and $\mathbb{T}$. So estimating HSIC for $(\mathcal{X}_s, \mathcal{Y}_s)$ and $(\mathcal{X}_t, \mathcal{Y}_t)$ can be carried out straightforwardly. However we have *no* direct sample from the joint distribution $\Pr_{st}$ because the samples in (3) and (4), *i.e.* the features from different domains, are not associated. Actually how to associate the features depends on the structures of each domain, and we therefore name the cross-domain dependency as "structural dependency", which can only be determined if we understand the structural analogy across the domains.

For a given association of the source and target domain features, as in (6), structural dependency between the domains can be estimated by (7). That means, by maximizing the estimated structural dependency, we find the "correct" association of the features from both domains, *i.e.* we make the *analogy* across domains.

Formally, given $W \leq \min(S,T)$, let σ_s and σ_t be injectives from $\{1, \cdots, W\}$ to $\{1, \cdots, S\}$ and $\{1, \cdots, T\}$ respectively, we could describe the learning problem as selecting a *ordered* set of features

$$\begin{aligned}\{f_{\sigma_s(1)}^{(s)}, f_{\sigma_s(2)}^{(s)}, \cdots, f_{\sigma_s(W)}^{(s)}\}, \text{ and} \\ \{f_{\sigma_t(1)}^{(t)}, f_{\sigma_t(2)}^{(t)}, \cdots, f_{\sigma_t(W)}^{(t)}\}\end{aligned} \quad (8)$$

from both the source and the target learning task, such that the objective function combining dependencies between $(\mathcal{X}_s, \mathcal{Y}_s)$, $(\mathcal{X}_t, \mathcal{Y}_t)$ and (Φ_s, Φ_t) is maximized:

$$\begin{aligned}(\hat{\sigma}_s, \hat{\sigma}_t) = \arg\max_{\sigma_s, \sigma_t}[\mathcal{D}(\mathcal{F}_s, \mathcal{F}_t, \mathbb{F}) \\ +\lambda_s\mathcal{D}(\mathcal{H}_s, \mathcal{G}_s, \mathbb{S}) + \lambda_t\mathcal{D}(\mathcal{H}_t, \mathcal{G}_t, \mathbb{T})]\end{aligned} \quad (9)$$

where $\mathbb{F} = \{(f_{\sigma_s(1)}^{(s)}, f_{\sigma_t(1)}^{(t)}), \cdots, (f_{\sigma_s(W)}^{(s)}, f_{\sigma_t(W)}^{(t)})\}$ is the pseudo-sample from the joint distribution $\Pr_{st}$ constructed by associating the selected features from both domains. All the three terms in (9) are estimated by the estimator (7) with kernel matrices $\mathbf{K}^s$, $\mathbf{K}^t$, $\mathbf{M}^s$, $\mathbf{L}^s$, $\mathbf{M}^t$ and $\mathbf{L}^t$ computed using the selected features in (8). λ_s and λ_t are free parameters controlling the relative influences the terms.

After determining σ_s and σ_t, each sample of the source domain can be "translated" into a sample for the target domain by treating the features $f_{\sigma_s(i)}^{(s)}$ and $f_{\sigma_t(i)}^{(t)}$ (analogues)

as equivalent. Then standard supervised learning methods can be applied to the expanded training set of the target domain. Computing the structural similarity between the domains also becomes straightforward. One can directly measure the structural similarity by $\mathcal{D}(\mathcal{F}_s,\ \mathcal{F}_t,\ \mathbb{F})$.

It is noticeable that the above described learning paradigm bears some key features that can be viewed as prototype models of the components in human's learning by analogy:

1. The learner knows the key concepts in a familiar case (source domain).
2. The learner identifies key concepts in a new problem (target domain) by both analyzing the new problem itself and making the analogy from a previous familiar case base on their structural similarities.
3. The learner gains better understanding of the new problem thanks to the knowledge transferred from the previous familiar case.

Algorithm

We have presented the general framework of learning by structural analogy. However, finding the globally optimal solution to the optimization problem in (9) is not straightforward. In this paper, we present a simple algorithm to implement the framework by finding a local minimum of the objective.

Our algorithm first selects features from both domains by maximizing $\mathcal{D}(\mathcal{H}_s,\ \mathcal{G}_s,\ \mathbb{S})$ and $\mathcal{D}(\mathcal{H}_t,\ \mathcal{G}_t,\ \mathbb{T})$ respectively, without considering relations between the two domains. Then we find the analogy by sorting the selected features for the source domain to maximize $\mathcal{D}(\mathcal{F}_s,\ \mathcal{F}_t,\ \mathbb{F})$. One advantage of this implementation is that we actually do not have to determine the weights λ_s and λ_t as the corresponding terms are maximized in separate procedures.

For feature selection, we simply sort all the features according the estimated HSIC (as in (7)) using the kernel matrix computed by only one feature. And then selected the top W features with largest estimated HSIC. This procedure ignores possible interactions between the features, but achieves better efficiency especially when dealing with large scale problems (such as the one in our real-world data experiment).

Then, sorting the selected features of the source domain to "make the analogy" is achieved by the algorithm proposed in (Quadrianto 2008). Specifically, we aim to find the optimal permutation π^* from the permutation group Π_W:

$$\pi^* = \arg\max_{\pi\in\Pi_W} \operatorname{tr} \bar{\mathbf{K}}^t \pi^\top \bar{\mathbf{K}}^s \pi \tag{10}$$

where $\bar{\mathbf{K}}^t = \mathrm{H}\mathbf{K}^t\mathrm{H}$, $\bar{\mathbf{K}}^s = \mathrm{H}\mathbf{K}^s\mathrm{H}$ and $\mathrm{H}_{ij} = \delta_{ij} - W^{-1}$. This optimization problem is solved iteratively by:

$$\pi_{i+1} = (1-\lambda)\pi_i + \lambda \arg\max_{\pi\in\Pi_W} \left[\operatorname{tr} \bar{\mathbf{K}}^t \pi^\top \bar{\mathbf{K}}^s \pi_i\right] \tag{11}$$

Since $\operatorname{tr} \bar{\mathbf{K}}^t \pi^\top \bar{\mathbf{K}}^s \pi_i = \operatorname{tr} \bar{\mathbf{K}}^s \pi_i \bar{\mathbf{K}}^t \pi^\top$, we end up solving a linear assignment problem (LAP) with the cost matrix $-\bar{\mathbf{K}}^s \pi_i \bar{\mathbf{K}}^t$. A very efficient solver of LAP can be found in (Cao 2008).

The whole procedure is formalized in Algorithm 1.

Algorithm 1 Transfer Learning by Structural Analogy

Input: $\mathbb{S}$ and $\mathbb{T}$.
Output: $\{f^{(s)}_{\sigma_s(1)}, f^{(s)}_{\sigma_s(2)}, \cdots, f^{(s)}_{\sigma_s(W)}\}$ and $\{f^{(t)}_{\sigma_t(1)}, f^{(t)}_{\sigma_t(2)}, \cdots, f^{(t)}_{\sigma_t(W)}\}$.
Compute $\mathbf{L}^s$ and $\mathbf{L}^t$;
for $i = 1$ **to** N_s **do**
  Compute $\mathbf{M}^s$ using only $f^{(s)}_i$;
  Estimate the HSIC $\mathcal{D}(\mathcal{H}_s,\ \mathcal{G}_s,\ \mathbb{S})$ using $\mathbf{M}^s$ and $\mathbf{L}^s$;
end for
Find W features from Φ_s with largest estimated HSIC;
for $i = 1$ **to** N_t **do**
  Compute $\mathbf{M}^t$ using only $f^{(t)}_i$;
  Estimate the HSIC $\mathcal{D}(\mathcal{H}_t,\ \mathcal{G}_t,\ \mathbb{T})$ using $\mathbf{M}^t$ and $\mathbf{L}^t$;
end for
Find W features from Φ_t with largest estimated HSIC;
Compute $\bar{\mathbf{K}}^s$ and $\bar{\mathbf{K}}^t$ with all selected features together;
Initialize permutation matrix π_0;
for $i = 0$ **to** $\mathrm{MAX} - 1$ **do**
  Compute cost matrix $-\bar{\mathbf{K}}^s \pi_{i-1} \bar{\mathbf{K}}^t$;
  Solve the LAP with the cost matrix;
  Update permutation matrix as in (11);
  if converged **then**
    break;
  end if
end for

Experiments

Ohsumed Dataset

We apply our method to the Ohsumed (Hersh 1994) text dataset[2]. The Ohsumed dataset consists of documents on medical issues covering 23 topics (classes) with ground truth labels on each document. The preprocessed corpus is bag-of-words data on a vocabulary of 30689 unique words (dimensions). We randomly picked 2 classes from the dataset, namely "Respiratory Tract Diseases" and "Cardiovascular Diseases". For each class we randomly sampled 200 positive examples and 200 negative examples, and we will try to automatically make analogy between these two domains, such that knowledge can be transferred for classification tasks.

We let $W = 10$ in our algorithm, and we automatically learned the top 10 words in each domain that are supposed to be "analogues" as in Table 1. We can see that the top 10 words selected from the two domains have almost no overlap, *i.e.*, they are in different low-level representations. However, the structural relatedness enables us to find analogues across domains. As we can see in Table 1, the automatically learned words indeed constitute several pairs of plausible analogues. For example, "infect" and "pneumonia" ("pneumonia" means infection in the lung); "valv" and "respiratori"; "cell" and "lung" ("cell" means an enclosed cavity in the heart); "aortic" and "tract" (they are both major passages in each sub-system of the body). Note that these analogues are automatically discovered without making use

[2]The dataset is downloaded from P.V. Gehler's page http://www.kyb.mpg.de/bs/people/pgehler/rap/index.html

Table 1: Learned analogy between the two domains

CARDI. DISEASES	RESPIRATORY TRACT DISEASES
ENDOCARD	INFECT
INFECT	PNEUMONIA
HEART	PULMONARI
VALV	RESPIRATORI
CELL	LUNG
COMPLIC	CULTUR
CARDIAC	BACTERI
AORTIC	TRACT
STUDI	CASE
EFFECT	INCREAS

of any co-occurrence information between the words from different domains.

To further justify the analogy found by our algorithm, we trained a linear classifier for the source domain documents, and applied it to the target domain documents by treating the *analogues* as equivalent. This procedure yields an accuracy of 80.50% on the target domain[3], which justified that the analogy found by our algorithm greatly helped in understanding the target domain.

Synthetic Learning Scenario

In order to further illustrate the importance of exploiting "structural" information within each domain in making the analogy. We show experimental results in a synthetic learning scenario.

Suppose that a learner is presented with a task of distinguishing two classes represented by two 10-dimensional Gaussian distributions with full covariance. The two classes are created such that they are hardly distinguishable in all the individual dimensions, but can largely be separated by a hyper-plane in the 10-dimensional space. For evaluation we hold out a test set of 2000 samples with known ground truth labels. The learner is required to learn a hyperplane to distinguish the two classes, without recourse to the test samples (standard inductive learning).

In our scenario, the learner only have 100 *unlabeled* samples in the training phase. With standard machine learning techniques, we first cluster the samples with the K-means algorithm, then train a classifier on the resulted two clusters (using standard linear discriminant analysis, *i.e.* fits a multivariate Gaussian density to each group, with a pooled estimate of covariance). The classifier obtained from this procedure yields an accuracy 72.35% on the test set.

The above scenario is quite difficult for traditional machine learning methodologies since the learner is only provided with a small number of *unlabeled* samples, which implicates that the learner have very limited understanding of the learning task. According to the *learning-by-analogy* philosophy, in such situations the learning should have recourse to a previous familiar case in order to solve the current problem. But the "previous familiar case" could have different representations from the current problem, which means the learner has to make an *analogy* across the two domains instead of directly copying the previous solution.

[3]A non-informative classifier would give an accuracy of 50% in this setting.

Table 2: Experiment results on the synthetic task

METHODS	ACCURACY ON TEST SET
WITHOUT PREVIOUS CASE	72.35%
INAPPROPRIATE ANALOGY	76.40%
OUR METHOD	**94.25%**

We synthesize such a "previous familiar case" by randomly permutating the 10 dimensions together with 10 additional noise dimensions which bear no information to distinguish the two classes. The learner is presented with 1000 labeled samples (500 for each class) from the 20-dimensional distribution, which indicates that the learner is quite familiar with this "previous case". However, such a "previous case" is of no use in traditional machine learning techniques as the feature space is different.

It is noticeable that there is rich "structural" information among the dimensions for us to exploit (as the data are generate from Gaussian distributions with *full* covariances). Specifically we apply our framework (let $W = 10$) to make an *analogy* between the 10 dimensions in the current problem and the 20 dimensions in the "previous familiar case". Note that the term $\mathcal{D}(\mathcal{H}_t,\ \mathcal{G}_t,\ \mathbb{T})$ vanishes as we have no labeled data for the target task, and $\mathbf{K}^t$ is estimated using the 100 unlabeled samples. After determined the "analogues" of the current problem's dimensions in the "previous familiar case", we translate the 1000 labeled samples to the current problem by treating the "analogues" as equivalent. We then apply standard linear discriminant analysis to the translated samples, and obtain a classifier for the current problem, which yields an accuracy of 94.25% on the test set.

Note that resolving the "structures" among the dimensions within each domain plays an essential role in successfully making the analogy. To verify this, we also tried to ignore the term $\mathcal{D}(\mathcal{F}_s,\ \mathcal{F}_t,\ \mathbb{F})$ and merely rank the dimensions according to their relevances to the label. In this way we obtain a classifier which yields an accuracy of 76.40%.

As summarized in Table 2, we can conclude that,

1. We cannot achieve satisfiable performance when we have limited understanding of the current problem and do not have recourse to previous cases.
2. We achieve little performance improvement if we have recourse to a previous familiar case but do not carefully analyze the structural of both domains and make an inappropriate analogy.
3. We finally achieve satisfiable understanding of the current problem through correctly making the analogy to a previous familiar case.

Conclusion

In this paper we addressed the problem of transfer learning by structural analogy between two domains with completely different low-level representations. By making use of statistical tools, we tried to bridge transfer learning and the old

paradigm of learning by analogy, and extend them to more general settings. The current work and our future research aim at automatically making structural analogies and determine the structural similarities with as few prior knowledge and background restrictions as possible.

Acknowledgement

We thank the support of Hong Kong RGC/NSFC project N_HKUST 624/09 and RGC project 621010.

References

Agnar Aamodt and Enric Plaza, Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, *Artificial Intelligence Communications* 7 : 1, 39-52. 1994.

D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, *Machine Learning*, 6, 37-66, 1991.

K.D. Ashley, Reasoning with Cases and Hypotheticals in Hypo, In *International Journal of Man?Machine Studies*, Vol. 34, pp. 753-796. Academic Press. New York. 1991

S. Bayoudh, L. Miclet, and A. Delhay, Learning by Analogy: a Classification Rule for Binary and Nominal Data. In *IJCAI*, 2007.

J. Blitzer, R. McDonald, F. Pereira, Domain Adaptation with Structural Correspondence Learning, In *EMNLP* 2006, pages 120-128,

Yi Cao, Munkres' Assignment Algorithm, Modified for Rectangular Matrices. http://csclab.murraystate.edu/bob.pilgrim/445/munkres.html

Jaime G. Carbonell, A Computational Model of Analogical Problem Solving, In *IJCAI*, 1981.

Jaime G. Carbonell, Oren Etzioni, Yolanda Gil, Robert Joseph, Craig A. Knoblock, Steven Minton, Manuela M. Veloso, PRODIGY: An Integrated Architecture for Planning and Learning, *SIGART Bulletin* 2(4): 51-55, 1991.

R. Caruana, Multitask learning, *Machine Learning*, 28(1):41-75, 1997.

W. Cheetham, K. Goebel, Appliance Call Center: A Successful Mixed-Initiative Case Study, *Artificial Intelligence Magazine*, Volume 28, No. 2, (2007). pp 89-100.

Wenyuan Dai, Yuqiang Chen, Gui-Rong Xue, Qiang Yang, and Yong Yu, Translated Learning: Transfer Learning across Different Feature Spaces. In *NIPS* 2008.

H. Daumé III, D. Marcu, Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26, 101C126. 2006.

Brian Falkenhainer, Kenneth D. Forbus, and Dedre Gentner The Structure-Mapping Engine: Algorithm and Examples *Artificial Intelligence*, 41, page 1-63, 1989.

K.D. Forbus, D. Gentner, A.B. Markman, and R.W. Ferguson, Analogy just looks like high-level perception: Why a domain-general approach to analogical mapping is right. *Journal of Experimental and Theoretical Artificial Intelligence*, 10(2), 231-257, 1998.

D. Gentner, The mechanisms of analogical reasoning. In J.W. Shavlik, T.G. Dietterich (eds), *Readings in Machine Learning*. Morgan Kaufmann, 1990.

Arthur Gretton, Olivier Bousquet, Alex Smola and Bernhard Schölkopf, Measuring Statistical Dependence with Hilbert-Schmidt Norms In *Algorithmic Learning Theory, 16th International Conference*. 2005.

Arthur Gretton, Kenji Fukumizu, Choon Hui Teo, Le Song, Bernhard Schölkopf and Alex Smola A Kernel Statistical Test of Independence In *NIPS* 2007.

William Hersh, Chris Buckley, TJ Leone, David Hickam, OHSUMED: An Interactive Retrieval Evaluation and New Large Test Collection for Research In *SIGIR* 1994.

K.J. Holyoak, and P. Thagard, The Analogical Mind. In *Case-Based Reasoning: An Overview* Ramon Lopez de Montaras and Enric Plaza, 1997.

Janet Kolodner, Reconstructive Memory: A Computer Model, *Cognitive Science* 7, (1983): 4.

Janet Kolodner, *Case-Based Reasoning*. San Mateo: Morgan Kaufmann, 1993.

D.B. Leake, *Case-Based Reasoning: Experiences, Lessons and Future Directions* MIT Press, 1996.

M. M. Hassan Mahmud and Sylvian R. Ray Transfer Learning using Kolmogorov Complexity: Basic Theory and Empirical Evaluations In *NIPS* 2007.

Bill Mark, Case-Based Reasoning for Autoclave Management, In *Proceedings of the Case-Based Reasoning Workshop* 1989.

D.W. Patterson, N. Rooney, and M. Galushka, Efficient similarity determination and case construction techniques for case-based reasoning, In S. Craw and A. Preece (eds): *ECCBR*, 2002, *LNAI 2416*, pp. 292-305.

N. Quadrianto, L. Song, A.J. Smola, Kernelized Sorting. In *NIPS* 2008.

B. Schölkopf, A.J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, 2001

A.J. Smola, A. Gretton, Le Song, and B. Schölkopf, A hilbert space embedding for distributions. In E. Takimoto, editor, *Algorithmic Learning Theory* , *Lecture Notes on Computer Science*. Springer, 2007.

Le Song, A.J. Smola, A. Gretton, K. Borgwardt, and J. Bedo, Supervised Feature Selection via Dependence Estimation. In *ICML* 2007.

Roger Schank, *Dynamic Memory: A Theory of Learning in Computers and People* New York: Cambridge University Press, 1982.

Sebastian Thrun and Lorien Pratt (eds), *Learning to Learn*, Kluwer Academic Publishers, 1998.

Ian Watson, *Applying Case-Based Reasoning: Techniques for Enterprise Systems.* Elsevier, 1997.

P.H. Winston, Learning and Reasoning by Analogy, *Communications of the ACM*, 23, pp. 689-703, 1980.